FINAL REPORT

ON

LANGUAGE UNDERSTANDING AND GENERATION OF COMPLEX TUTORIAL DIALOGUES*

Martha Evens
Department of Computer Science
Illinois Institute of Technology
10 West 31st Street
Chicago, IL 60616
evens@iit.edu
312-567-5153 or 847-869-8537

* This work was supported by the Cognitive Science Program, Office of Naval Research under Grant No. N00014-94-1-0338, to Illinois Institute of Technology. The content does not reflect the position of policy of the government and no official endorsement should be inferred.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT DATE	3. DATES COVERED *(From - To)*
30-09-2000	final report	Nov 1994-Sep 2000

4. TITLE AND SUBTITLE

Language Understanding and Generation in Complex Tutorial Dialogues

5a. CONTRACT NUMBER

5b. GRANT NUMBER

N00014-94-1-0338

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)

Evens, Martha W.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Computer Science Department
Illinois Institute of Technology
10 West 31st Street, Chicago, IL 60616

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Cognitive Science Program
Office of Naval Research
800 North Quincy
Arlington, VA 22217-5000

10. SPONSOR/MONITOR'S ACRONYM(S)

ONR

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT

UU

13. SUPPLEMENTARY NOTES

14. ABSTRACT

CIRCSIM-Tutor is an intelligent tutoring system for the domain of cardiovascular physiology, which carries out a natural language dialogue with the user, using a set of tutoring tactics that mimic those employed by expert human tutors. The system has been used extensively by students at Rush Medical College, with positive learning outcomes. The system manages to understand and respond to more than 98% of student inputs using advanced methods of spelling correction and information extraction. Studies of human tutoring sessions have revealed the frequency and variey of hints, and the way they are intertwined with acknowledgments. SGML markup and machine learning have allowed us to classify student initiatives and tutor responses and generate rules for deploying hints and other sophisticated strategies. Comparison of novice and expert tutors has revealed striking differences.

15. SUBJECT TERMS

Intelligent tutoring systems, natural language generation, dialogue schemas, tutoring strategies, discourse analysis, understanding ill-formed input, SGML markup, hints, mixed initiative dialogue

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPONE NUMBER *(Include area code)*
					Martha W. Evens
U	U	U	UU	35	312-567-5153

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI-Std Z39-18

FINAL REPORT OF THE CIRCSIM-TUTOR PROJECT GRANT N00014-94-1-0338 LANGUAGE UNDERSTANDING AND GENERATION IN COMPLEX TUTORIAL DIALOGUES FOR THE PERIOD FROM Nov 1994 to Sep 2000

1 Introduction

CIRCSIM-Tutor is an intelligent tutoring system for the domain of cardiovascular physiology, which carries out a natural language dialogue with the user, using a set of tutoring tactics that mimic those employed by two expert human tutors. CIRCSIM-Tutor has been used extensively by students at Rush Medical College, and the learning outcomes of a one hour interaction with the program have been demonstrated. Student reaction to the program was very positive. Here we describe the design and development of CIRCSIM-Tutor and the studies of human tutoring upon which that design is based.

We begin by describing the current behavior of our system in trials with medical students. Then we discuss the collection and analysis of human tutoring dialogues, which form the basis of our system, and how novice tutoring strategies compare with those of experts. Then we describe the development of our knowledge base, our experiments with different planners, our attempts at modeling the student/user. Finally we discuss the approaches to natural language understanding and generation that we have implemented in our system.

The Circsim-Tutor project grew out of a unique collaboration between experts in physiology, who are also experts in tutoring, and experts in natural language. It started because years of experience building computer-based learning systems for their students convinced Joel Michael and Allen Rovick, Professors of Physiology at Rush Medical College, that real natural language interaction was essential to building better systems. They had developed a methodology for teaching physiology in a manner that fosters the development of problem solving skills by medical students, specifically through the use of (1) tutorial interactions (both one-on-one and small group), and (2) simulation-based computer-assisted instruction (CAI). They had been building CAI systems for years, including a PLATO program called HEARTSIM (Rovick & Brenner, 1983) and CIRCSIM (Rovick & Michael, 1986, 1992).

At the same time they had become increasingly aware of how much time and attention they devote to the use of language in their small-group and one-on-one sessions with students. They were convinced that teaching the language and the content of their discipline are inextricably intertwined. Language, they felt, must be an integral part of any tutoring dialogue that tries to give students a high level of understanding of complex processes.

The development of simulation-based CAI culminated in a program called CIRCSIM (Rovick & Michael, 1986). The educational objective of CIRCSIM is to assist the students to develop a coherent mental model of a particular negative feedback system, and to learn a problem solving process for predicting the behavior of this system and others like it. CIRCSIM presents students with a problem, a perturbation to the negative feedback system that acts to stabilize the blood pressure. Then it asks the student to predict the qualitative behavior of seven important physiological parameters in response to this perturbation. It analyzes these predictions, identifies the errors, links them to the most likely underlying misconceptions, chooses a canned paragraph of textual explanation to remedy them from over 240 alternatives, and presents that paragraph to the student.

CIRCSIM is an effective learning resource; it is well received by students and it has been shown to have an appreciable impact on their learning (Rovick & Michael, 1992). Still, CIRCSIM is a conventional CAI program. It lacks the ability to understand or generate natural language text. Hence, it accepts only simple key strokes as inputs and can only present stored text in tutoring or in offering explanations. Furthermore, its student model is "generic" not student specific.

As Michael and Rovick searched for ways to improve CIRCSIM, they became more and more frustrated by its inability to carry on a true natural language dialogue between students and the computer. They realized that the "canned" output significantly limited the kinds of interactions, evaluations, and instruction that a program can deliver. They became convinced that the lack of natural language dialogue severely limited the ability to uncover misconceptions and remedy them. Their search for natural language expertise brought us together. I was impressed by their expertise in CAI and their ideas about tutoring and eager to try to dialogue generation.

We set out to develop the ability to understand and generate the language of cardiovascular physiology on the computer. Obviously, we needed to develop a knowledge base for this complex and domain. Even more we needed to find out how to organize and structure the tutoring session and to discover how to carry on a tutoring dialogue. To answer all these questions we turned to the study of actual tutoring dialogues. We also determined that the tutor we built should fit into course laboratories but also be designed to run on its own for students who wanted to review this material in preparation for boards. We decided to plan the implementation in incremental style so that it could be tested with actual students repeatedly during development.

Nakhoon Kim (Kim et al., 1989) wrote a Prolog prototype so that we could begin to understand the collection and analysis of student predictions and build a first version of the student model. ONR was using Xerox Lisp machines at that point so we borrowed two from the ONR center at LRDC and got them up and running with the help of Alan Lesgold. Yoon Hee Lee wrote the first input understander on one of these machines while Jun Li wrote the first screen manager and Yuemei Zhang (1991) wrote the first generation program on the other. These pieces of "Version 1" were never integrated, because Xerox went out of the hardware business and ONR decided to switch to the Macintosh. We converted these programs to Procyon Common Lisp. Chong Woo wrote a planner and integrated the pieces into Version 2 with a student modeler from Leemseop Shim and a new generator from Ru-Charn Chang (Woo, 1992). See Figure 1 for a system diagram for Version 2 and Figure 2 for a screen print. The system presents a problem situation and asks the student to predict the qualitative behavior of seven important variables (shown on the lower right of the screen). Then it marks the prediction errors with a slash across the box and starts a remedial dialogue with the student.

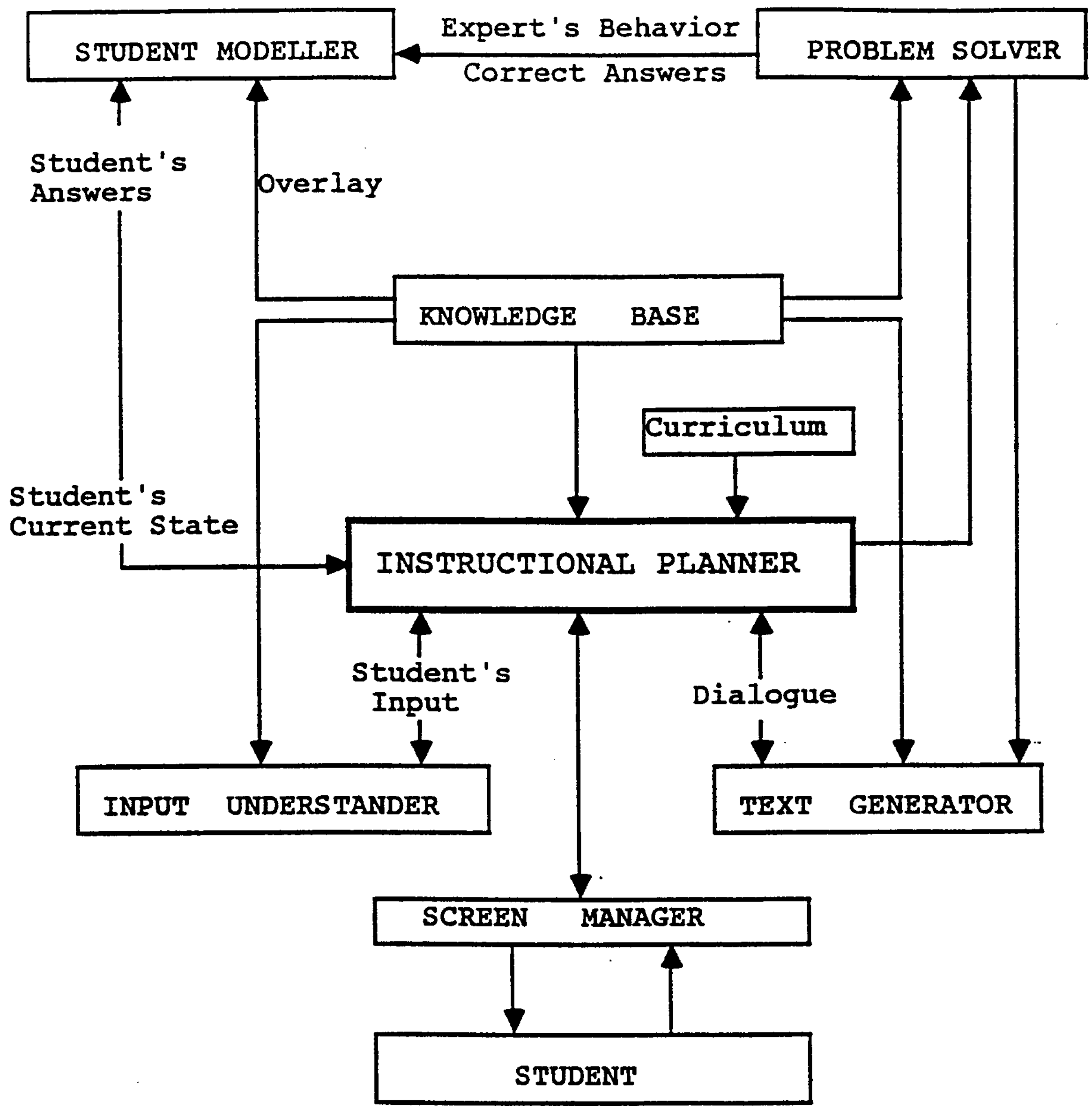

Figure 1. CIRCSIM-Tutor System Diagram for Version 2
(Woo, 1992, p. 9)

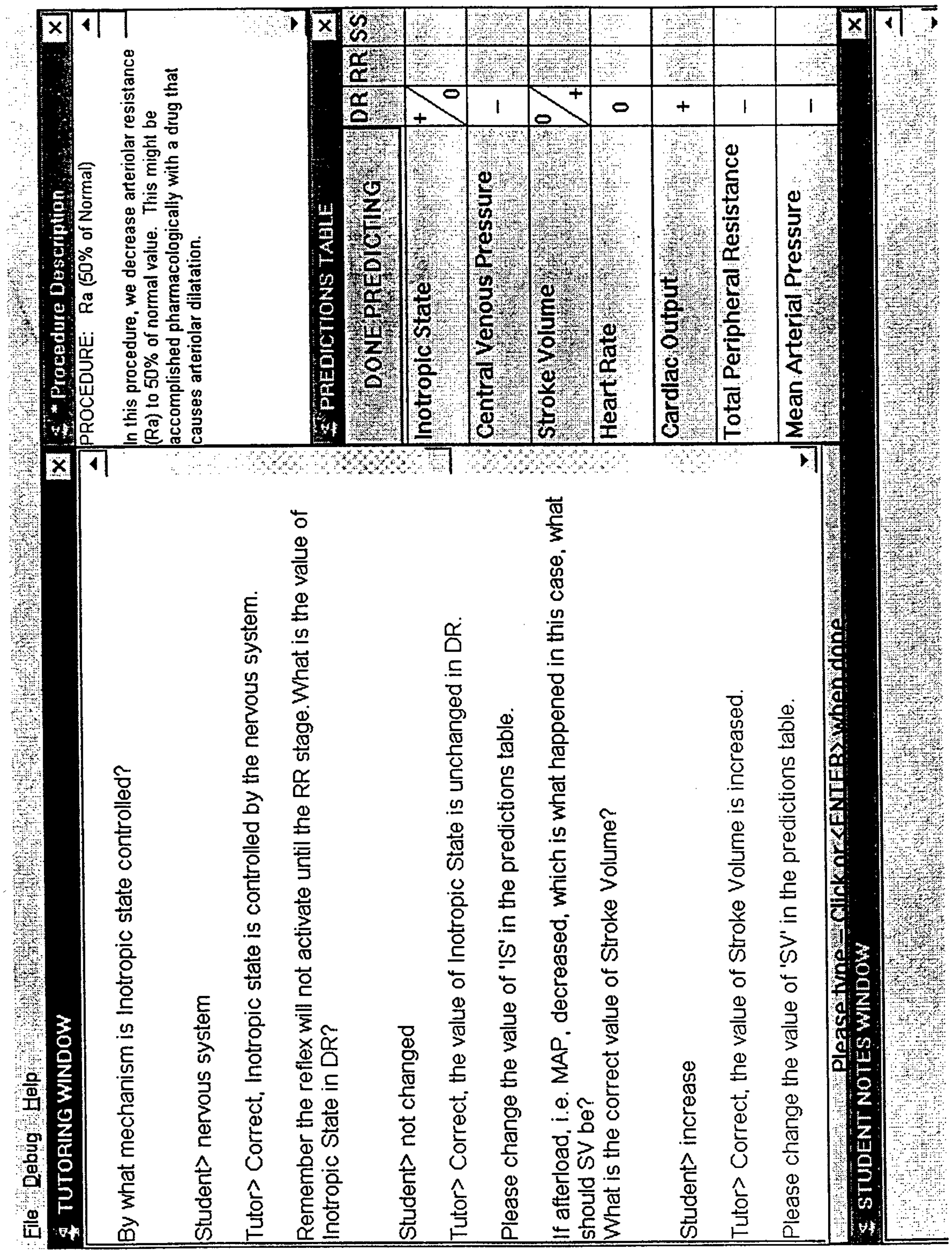

Figure 2. Screen Print from CIRCSIM-Tutor Version 2.9 (November, 1999)

We paid medical students to use the system one at a time just as soon as we put it together in 1992. We learned a tremendous amount even from those first encounters. Michael and Rovick complained that the hints were terrible and we began to realize that we had misunderstood what they said about hints. They also complained when the system reused the same tutoring strategy that had just failed with a student. We realized that we needed to add tutoring history files and alternate strategies. We began to read papers on discourse analysis from psychology, sociology, and computer science.

It took five years of work (1993-1998) until the system was ready to be used as a regular part of the course. As we studied the tutoring transcripts we realized that the multiturn strategies and complex language and different tutoring protocols (Khuwaja, 1994) that we saw would be hard to implement with the generation patterns implemented one sentence at a time in Version 2. We started to talk right then about the Version 3 that is just coming to life now in 2000.

A major part of the effort was improvement of the input understanding programs – students are not comfortable carrying on a dialogue with a program that does not seem to understand them. The original input understanding program responded far too often with "I am sorry, I do not understand you. Please rephrase." Finally, Michael Glass (1999) wrote a totally new input understanding program using an information extraction technique. In April, 1998, twenty-three first-year students from the alternative (problem-oriented) curriculum at Rush used our system in a one-hour laboratory session. To our surprise many of them completed three procedures. These students expressed a great deal of enthusiasm in a survey we administered immediately after their session; they also made a number of very useful suggestions.

2 Medical students are using CIRCSIM-Tutor and it works

In November, 1998, fifty students tried the system in a two hour laboratory with pre-tests and post-tests and questionnaires. The students showed marked improvement from pre-test to post-test. What is more, the improvement appeared not just on CIRCSIM-like problems but carried over to different kinds of reasoning problems. Twenty-four students used the system in pairs and twenty-six were alone at a machine. The advantage shown in working with a partner in the CIRCSIM experiments did not appear here. This time the majority of the students completed four procedures. Their answers on questionnaires appeared to be even more enthusiastic.

What is more, the system did not fail under the onslaught and the new input understander was able to respond to over 95% of the inputs. When the system did not understand the student, it described the kind of input it was expecting in terms the student could understand. In the following example the student gives an answer that the system does not understand, but it manages to respond appropriately anyway. More examples of Version 2 delivering hints appear in Figure 5.

Tutor: By what mechanism is Total Peripheral Resistance controlled?

Student: Metabolic factors

Tutor: Is the mechanism of control neural or physical?

Student: Neural

This success persuaded us to try a full-scale experiment in November, 1999. Yujian Zhou rewrote the basic student modeling program and added three more modeling components in order to tailor hints and acknowledgements to the user (Zhou 1999a,b). Also, Byung-In Cho built a curriculum planner that uses a global assessment of student performance to plan an individual curriculum for each student (Cho et al. 1999).

Our experiment in November, 1999, involved all the students in the first-year class at Rush Medical College. Half of the class used CIRCSIM and the other half used CIRCSIM-Tutor. Twenty-five of them were also tutored in keyboard-to-keyboard fashion by Michael and Rovick the weekend before the laboratory sessions. A control group read a prepared text and recevied no tutoring at all. The students took pre-tests and post-test and also filled out questionnaires designed to discover how the students reacted to both computer-based tutoring systems. The analysis of the results is not complete, but again we see significant improvement in students using CIRCSIM-Tutor.

3 Collection and analysis of human tutoring dialogues

The collection and analysis of human tutoring dialogues has been the basis of the design of our system every step of the way. We have now collected seventy-five transcripts of keyboard-to- keyboard tutoring sessions, mostly one hour long, carried out with the student in one room and our expert tutor in another, with the goal of capturing the kind of dialogue we wanted the machine tutor to produce. Li (1992b) wrote the CDS system to allow us to capture these dialogues. During the last year Zhou rewrote CDS in C++ so that it uses the Internet. The expert tutors were Michael and Rovick, who are domain experts in physiology and pedagogical experts in tutoring. We have also collected thirty tutoring sessions carried out by novice tutors (Glass et al., 1999).

Most of these sessions have lasted one hour or close to it, during which the tutor and the student solved one problem together. Most earlier studies of human tutoring had been carried out with grade school or high school level students, where poor motivation and poor performance were the major issues. Impressed by the tutoring skills displayed by our experts, Ramzan Ali Khuwaja (Khuwaja, 1994) suggested that we carry out a series of two hour experiments, in the hope that we might see changes in the student language and improvement in their problem-solving skills. Rovick and Michael did indeed carry out nine such two hour sessions, with the students working on two different problems. At Khuwaja's suggestion we also arranged for a control group of medical students to read some selected materials and take the same pre-test and post-test as the tutored students. Results showed that the improvement seen in the tutored students was significantly greater than the improvement shown in the control group even with this small number of students. Thus tutoring produced a demonstrable improvement even in these highly intelligent and thoroughly motivated students (Michael & Rovick, 1993).

Analysis of these sessions is the basis of the tutoring strategies and tutoring tactics, the problem-solving components, the student modeler, and the domain knowledge base in CIRCSIM-Tutor. We applied the same kind of approach to tutoring language as we set out to discover how a tutor generates the language that represents one side of a tutorial dialogue. We analyzed the transcripts to determine what the tutor chooses to talk about and how that information is organized and expressed.

When we began to analyze the transcripts, we had very little experience in dialogue analysis, so initially most of our analysis was intuitive. We read and reread transcripts and tried to express what we saw in the form of rules. We also asked our expert tutors question after question. After each student turn we asked: How does this answer change your ideas about the student? What do you think is the source of confusion here? After each tutor turn we asked: Why did you ask that question? What are you trying to accomplish here? Sometimes the tutors could tell us; sometimes they could not. Many times we asked the wrong questions. A visit from Kurt VanLehn taught us that it is much more effective to ask questions while the tutoring is in progress and also launched Hume's investigation of hinting.

Almost unnoticed, Hume also took an important methodological step. He entered the hint categories into the electronic version of the transcript. At the urging of Reva Freedman, we started to place SGML markup in our transcripts to describe all the phenomena we saw. She put us in touch with the dialogue markup carried out by Allen and Moore (DAMSL, 1997) on task-assistance dialogues. SGML markup allowed us to make much more accurate counts of various phenomena. The distribution of free SGML tools from Edinburgh (McKelvie et al. 1997) allowed us to automate this counting process and also to apply machine learning programs much more easily. Zhou was the first among us to apply machine-learning techniques to our transcripts. Often the output of the machine learning process is rules that we initially intuited, but sometimes new and better rules drop out (Freedman et al., 1998a,b; Kim et al., 1998a,b). The markup process is very labor intensive but the output has justified the effort. Kim's (1999) markup manual was an important step in making our markup consistent and repeatable. (An example from this manual is shown in Figure 3.) This work also makes it easier for us to revisit old questions and substantiate the results with statistical analysis.

4 Experiments with novice tutors

We undertook two sets of experiments with novice tutors, one in 1994 and the other in 1996, with the goal of trying to characterize expertise in tutoring (Glass et al., 1999). The sixten transcripts from the first experiment showed so many tutoring errors in physiology and so many problems in using CDS that we decided to try again and the analysis reported comes from the second set of fourteen transcripts.

There are major differences between the novice tutors and the experts. Most important, the experts are much more likely to give hints and ask questions, where the novice tutors tell the

```
<T-tutors-via-determinants  var=RAP>
 <T-tutors-determinant>
  <T-elicits>
          tu:What parameter determines RAP?
   <S-ans  catg=near-miss>
          st:CVP.
   </S-ans>
  </T-elicits>

  <T-moves-toward-PT  method-type=inner>
   <T-tutors-determinant  var=CVP>
    <T-elicits>
          tu:What determines CVP?
     <S-ans  catg=near-miss>
          st:Blood volume [CBV].
     </S-ans>
    </T-elicits>
   </T-tutors-determinant>
  </T-moves-toward-PT>

  <T-moves-toward-PT>
   <T-tutors-determinant  var=CBV>
    <T-elicits>
          tu:What determines CBV?
     <S-ans  catg=correct>
          st:CO.
     </S-ans>
    </T-elicits>
   </T-tutors-determinant>
  </T-moves-toward-PT>
 </T-tutors-determinant>

 <T-tutors-value>
  <T-elicits>
          tu: How would RAP change?
   <S-ans  catg=correct>
          st: Decrease.
   </S-ans>
   <T-ack  type=positive>
          tu: Correct.
   </T-ack>
  </T-elicits>
 </T-tutors-value>
</T-tutors-via-determinant>
```

Figure 3. Example from Jung Hee Kim's SGML Markup Manual Showing Mark-up of the Response to a "near miss" Student Answer (1999).

students the answer. If we look at which participant states the final value for the variable in a series of DR tutoring episodes, we see that the expert tutors get the student to give that value 85% of the time, while the novice tutors get the student to give the value only 56% of the time. The novice tutors also drag in extraneous concepts (Kim, 2000); they consistently use more concepts in tutoring a given variable than the experts do. The experts are more successful in getting active participation from the students. On the average, there are 4.74 student initiatives per session in the expert sessions and 3.21 in the novice sessions.

The novice tutors also ask the students "Do you understand?" or "Right?" while the experts almost never ask such questions; they ask substantive follow-up questions instead. Michael and Rovick told us at the beginning of our work together that CIRCSIM-Tutor should never ask such questions. They had already discovered what Graesser (1993a,b) has since demonstrated more formally; such questions are a waste of time.

5 Building the knowledge base and the problem solver

The existing Knowledge Base is the sixth that we have built to support Circsim-Tutor. These changes in the Knowledge Base have come about because of new requirements from the Problem Solver and from the generation components.

As our understanding of the complexity of the generation task has increased we have discarded the old problem solver and built a new problem-solver and a new knowledge base five times. Nakhoon Kim (1989) built the first one as part of a Prolog prototype. That first problem solver solved all the problems correctly, but not in the way that Allen Rovick and Joel Michael wanted to teach the students to solve them. The knowledge base was a collection of Prolog rules. So Kim built a second problem solver and rewrote the knowledge base to support it. This one solved the problems the way the tutors wanted the students to learn to solve them. But this problem-solver still did not provide a trace of the problem-solving process that the machine tutor could use as a basis for tutoring. Kim (1989) replaced it with a forest, a set of trees, one for each of the four procedures in the system at that time; each tree represented the ideal solution path for that procedure.

Yuemei Zhang (1991), who wrote the initial version of the text generation component in Lisp, was still not satisfied. She complained that the solution paths did not give her a representation of the problem-solving process that she could describe to students (1987). She built a new knowledge base, a frame system that represents the problem-solving algorithm in a declarative form, as well as all the concept map information. Zhang (1991) also pointed out the need for some higher-level concepts not originally represented in the knowledge base like "neural variable," so that the tutor can explain that "neural variables don't change in DR." This frame system is still in use in Version 2 with some additions from Yujian Zhou to support four more procedures and her new student model (Zhou, 2000).

The need for a new knowledge base for Version 3 was demonstrated by Ramzan Ali Khuwaja (1994). He envisioned a three layer knowledge base with many more procedures and a curriculum planning component to manage them and then implemented this knowledge base in CLOS. He also persuaded Allen Rovick to write more procedures and to develop procedure descriptions at different levels of complexity for use as the students progressed in sophistication.

Reva Freedman argued for replacing much of the knowledge embedded in frames by rules, which are easier to understand, easier to change, and easier to write about. She actually carried out the difficult task of representing the tutoring strategies and tactics in this way in her dissertation (1996). Increases in speed and memory size over the last ten years have made it possible to interpret rules in real time.

The task of developing and testing the rules for curriculum planning as well as adding the rules to support 83 procedures and procedure combinations has actually been carried out in the last year by Byung-In Cho (Cho et al., 2000a, 2000b). He has written the curriculum planner as a set of planning operators in Freedman's new Atlas Planning Environment (2000a,b), described in the next section.

6 Planning as a central issue in the generation of tutorial dialogues

The more we studied the tutoring transcripts the more we came to realize the tremendous amount of planning that expert tutors actually accomplish. They may plan what procedure to present in advance, but most of that planning is done dynamically, during the tutoring dialogue. The tutor plans to discover the student's misconceptions (the prediction table is major help here) and then plans to remediate those misconceptions. The remediation strategy typically takes several steps, each with its own set of alternative tactics. Then the tutor must plan how to deliver each message in sentences that themselves require further planning.

One of the major achievements of our research project was the planner built by Chong Woo Woo (1991) to solve these problems. It is a dynamic hierarchical planner that supports multiple layers of goals and subgoals in the lesson planning process and then multiple layers of strategies and tactics to carry them out these plans. Woo not only designed and built the planner but he integrated the natural language components, the problem-solver, the knowledge base, and the student modeler into a functioning system with the planner as controller (Woo, 1992).

Woo's planner is still the central component of Version 2, where it has driven the system through all our trials with medical students. It has continued to support the system through multiple changes in other components, but over the years some problems have been noted. Sanders (1995) described several kinds of multiturn tutoring strategies carried out by the expert tutors, such as multistep hints and directed lines of reasoning; he suggests that it would be easier to implement these strategies if we separated the planner and the control functions that are combined in Woo's design.

Freedman (1996) pointed out the problems that occur when a student unexpectedly fulfills several tutorial goals in one turn. Suppose the system asks the student for the determinant of cardiac output and the student not only tells us that the determinant is stroke volume, but also informs us that since the stroke volume has gone up, the cardiac output must go up as well. The system sounds very stupid, if it goes ahead and asks the student for the relationship between the variables and then for the change in cardiac output. But the code required to recognize what has occurred and fix the plan is very messy. She showed that a planner that checks at every step whether its goals have been satisfied can behave much more like a human tutor. She also noted that the central structure of Woo's system is to lay out a lesson plan and follow it, but that the expert tutors have as an even higher level goal the need to sustain the dialogue.

Freedman (2000a,b) has now developed a reactive planner ATLAS, that can handle these problems while carrying out multilevel plans, and that provides in the Atlas Planning Environment (APE) a way to integrate tutorial planning and discourse planning. This work was carried out at the University of Pittsburgh as part of Kurt VanLehn's CIRCLE project, where it serves as the engine for the ATLAS tutor. Atlas was motivated not only by difficulties with Woo's planner but by problems identified in using the Longbow text planner of Young and Moore (Young, 1994), which in turn was based on UC-POP.

In reactive planning the system chooses a schema for a new dialogue segment, but does not produce a detailed plan for the next turn until it proceeses the student response. Reactive planning corresponds well to the needs of tutorial dialogue. There is no need to plan the whole dialogue in detail, because the system cannot predict how the student will respond. However, the system may choose a multiturn schema to deliver a summary or to remediate a misconception. This schema serves as a top-level outline for a discourse segment. After each student response, the system decides whether to continue with the current schema, to insert some extra material before proceeding, or occasionally to abandon the schema because the student has revealed signs of deep confusion. The APE approach avoids the need to backtrack, which is essentially impossible in a conversation - the system cannot un-say a previous remark because the student did not give the expected response.

The ATLAS user must produce operators that contain goals for each task the planner is trying to accomplish. Each operator contains goals, as well as preconditions that must be satisfied before the operation can be performed, a set of steps for the operation, and a filter, which is a list of Well-Formed Formulas that must be in the database before the system runs the operator (Freedman, 2000a,b).

Our Version 3 is now written in APE as well. This decision required us to rewrite parts of the user interface and the problem solver, but the result is a much cleaner design and much more readable code.

There are two important differences between the architecture of Version 2 and the architecture of Version 3. Version 3 has more knowledge stores (in addition to the Domain Knowledge Base and the Student Model, there is a Lexicon, a Tutoring History, and a Dialogue History) and they can be accessed by any module in the system. The box marked Instructional Planner in the Version 2 diagram has been replaced by three boxes in Version 3 (the Curriculum Planner, the Discourse Planner, and the Turn Planner), while the Text Generator box is now a surface realization engine. In fact, the planning is all done by Freedman's Atlas Planning Engine and these boxes are separate sets of Atlas planning operators.

7 Modeling the student

The original student model, designed by Leemseop Shim, was basically an overlay model plus a list of known misconceptions, a very primitive buggy model. We stored c's and w's (c for correct and w for wrong) to record each answer given by the student in the prediction table or in the natural language tutorial dialogue.

We decided that we needed a certainty function defined for strings of c's and w's. We wanted it to take values in the range [0,1] so that we can compare them with probability values if we can ever figure out a way to establish valid probability estimates. More important, this function needed to model the conviction of our expert tutors that the most recent evaluation of the student's response is the most important. Thus, if the leftmost character in the string is the oldest and the rightmost is the newest, we require that

$CF(c) > CF(w)$

$CF(cc) > CF(wc) > CF(cw) > CF(ww)$

$CF(ccc) > CF(wcc) > CF(cwc) > CF(ccw) >$

$CF(wwc) > CF(wcw) > CF(cww) > CF(www)$

Our tutors feel that three responses on a given topic is the most that they ever remember, so at the moment we use only the three most recent responses. It seems reasonable also to set the value to 0.5 for the empty string. In other words, before we receive any information we have an estimate of certainty of 0.5. We decided to use finite convolutions to model this behavior. Thus we define our certainty function as follows:

$$CF(R_1, \ldots, R_n) = \frac{R_{n-k+i}W_{n-k+i} + \ldots + R_{n-1}W_{n-1} + R_n W_n}{W_{n-k+i} + \ldots + W_{n-1} + W_n}$$

where $CF(R_1, \ldots, R_n)$ is the value after n responses,
R_n is the nth response,
k is the window size, i.e., the number of responses considered,
W_n is the weight for the nth response,
$R_{n-k+i} = 1.0$ if the response is "c,"
$R_{n-k+i} = 0.0$ if the response is "w,"
and if $n - k + i < 1$, then $R_{n-k+i} = 0.5$, for an unknown value.

If the weights $W_n, W_{n-1}, \ldots$ are set to non-negative values and at least one is nonzero, it follows that the value will always be defined and will always lie in the unit interval. This brings us to the question of how to choose the value of k and the last k weights. Since our tutors claim they never remember more than three answers on any topic, we have temporarily set $k = 3$.
Our tutors seem comfortable with the weights:

$$W_n = 5, W_{n-1} = 3, W_{n-2} = 1.$$

Thus we accept that the student knows the concept if the value of the certainty factor is .95 or greater and that there is serious confusion if the value is .5 or less.

This formula has several advantages: it is easy and fast to compute; it obviously weights more recent information more heavily; and it does not require that the model be initiatlized to some preset value.

Our expert tutors found it very difficult to discuss student modeling issues. Apparently this part of the tutoring task is less conscious than hinting, for example. They do describe using both global and local assessments as the bases for their choice of hints and acknowledgments. Since Shim's model does not provide this kind of assessment, Yujian Zhou decided to redesign and rebuild the student model to overcome these limitations.

Zhou's model contains four components, designed to provide input for four different levels of planning: the global assessment (an overall assessment of the student's performance), the procedure-level assessment (an assessment of how the student is performing on this procedure so far), the stage assessment (one for each stage, DR, RR, and SS), and the local assessment (measured for each variable that has been tutored in this stage).

The global assessment combines an assessment of how well the student is performing in making initial predictions in the prediction table, how well the student responds to hints, and how well the student is doing in the tutoring dialogue. The procedure assessment contains these same variables looked at only in the current procedure, etc. Each answer is categorized as correct, partially correct, a near miss, an "I don't know" answer, or totally wrong, and this answer category is recorded in the tutoring history and weighted to produce a performance score (Zhou, 2000).

The student model is still not storing any measure of student unease. We are convinced that, when the student makes angry remarks or indicates uncertainty, the system should notice this and try to relieve the student's frustration.

8 Understanding natural language and spelling correction

The original language understanding program for CIRCSIM-Tutor was a simple bottom-up chart parser written by Yoon Hee Lee (Lee & Evens, 1998) on a Xerox Lisp machine. Lee spent most of his time and energy on spelling correction because he felt that this was the real challenge for a system accepting free natural language input. At the time I tried to convince him to work on the parser instead, but I am now convinced that he was right. The problem of spelling correction in a dialogue system is very different from the word processing applications that most people are familiar with. Students do not want to choose between alternative spellings; they want the system to figure out what they mean and continue on with the dialogue as a human tutor does. The students used a lot of medical abbreviations, which Lee added to the lexicon along with error forms too small to recognize by standard correction algorithms like "teh" and "hte" and "fo." They also invented spontaneous abbreviations quite often by stopping typing part of the way through a word. Lee handled this by reducing error cost for missing letters as the system got closer to the end of a word.

At the beginning of this project we carried out extensive studies of the language used by the tutor and by the student (Seu et al., 1991) and then built a lexicon using tools developed for the IITLEX project by Ahlswede, Conlon, and Strutz (Ahlswede, 1985; Conlon et al., 1993, 1994). Dardaine (1996) wrote case frames for use in parsing and generation.

In order to deal with some of the ambiguities that Lee's parser could not handle, Elmi (1994) wrote a new top-down parser, which proved to be too slow for our application but which works quite well on newspaper text. In the process (Elmi & Evens, 1998), he reprogrammed and speeded up the spelling correction algorithms and this part of his work survives in the current system. C.P. Rosé has also adopted our approach to spelling correction in the LCFLEX parser, which is being used in other tutors.

Michael Glass (1996, 1997, 1999) developed a new understander, modeled on the technology developed for information extraction. The central mechanism is a cascade of finite state transducers. Finite state machines are popular because they are fast and modular (Roche and Schabes, 1997). Each machine produces an output, which is usually some modification of the input.

The new module has a number of special purpose finite state machines. One FSM copes with copula deletion, removing finite forms of the verb "to be," but leaving the abbreviation "is" for inotropic state. Another looks for names of parameters and their abbreviations, and also verbs of change. Another looks for negations and combines them with verbs of change, so that "doesn't change" is tranformed into "**neg + change.**" Another looks for words and phrases that indicate proportionality.

This new module did exceptionally well in the experiment in November, 1998. Out of 1801 student turns, only 24 were not understood. Ten of these were so garbled or ambiguous that humans could not understand what the student meant either. Another nineteen made sense but were not recognized in any useful manner. Six of these nineteen had spelling errors that the system could not correct (but it did correct thirty such errors appropriately). In seven cases the system failed because of a missing or incomplete lexical entry (two of these involved abbreviations). In two cases the student asked for help but the system did not understand. Two more turns included unprintable expressions of frustration. Finally, two involved domain concepts beyond Circsim-Tutor's knowledge.

9 Generating natural language dialogues

9.1 Multiturn planning and directed lines of reasoning

Gregory Sanders (1995) was the first to recognize and study the many places where Michael and Rovick show evidence of plans that involve a long series of turns. He first noticed this phenomenon in the following summary, which he called a "Directed Line of Reasoning" or DLR, for short (Sanders, 1995, p.94).

K12-tu-65-2: Now consi.e. the first things that are going to change are the things that are under neural control, which of these determinants would be the first affected?

K12-st-66-1: Cc

K12-tu-67-1: Of course!
K12-tu-67-2: And in what direction?
K12-st-68-1: Decrease
K12-tu-69-1: Rightr again.
K12-tu-69-2: And how would that affect SV?
K12-st-70-1: Decrease
K12-tu-71-1: Sure.
K12-tu-71-2: And what affect would that have?
K12-st-72-1: Decrease co
K12-tu-73-1: Yes again.
K12-tu-73-2: Then what?
K12-st-74-1: Map d
K12-tu-75-1: Yes, again.
K12-tu-75-2: And in this regard.
K12-tu-75-3: It is MAP that is regulated by the BAROceptor reflex.
K12-tu-75-4: That's why it's called that.

He started to look for more examples and realized that shorter ones occur quite frequently in Michael and Rovick's tutoring sessions. When they want to produce an explanation or deliver a summary or remediate a student misconception, they typically do so in as interactive a manner as possible. Students often confuse Cardiac Contractility with the Frank-Starling (length-tension) effect. The tutors have developed a plan for remediating this misconception):

Step 1. Describe the Frank-Starling efffect.
Step 2. Define Cardiac Contractility
Step 3. Explain the relationship between them.

Apparently, when they are executing such a plan, they decide at each step whether the student might already possess this piece of information. If so, they ask the student; if not, they provide it themselves. Thus an implementation of this plan may look like this (Sanders, 1995, p. 90), if the student seems totally confused:

> You are confusing the Frank-Starling effect with IS. They are not the same. You will recall that the Frank-Starling effect is a length-tension relationship of muscle fibers. An increase in filling or preload (EDV) results in an increase in SV. In contrast, IS is determined by the autonomic nervous system. A change in IS will cause a change in SV with EDV held constant. In effect, an change in IS (a positive inotropic effect) will shift the Frank-Starling curve along the axis.

but like this if the student is otherwise doing well:

T: You are confusing the Frank-Starling effect with IS. Do you recall the Frank-Starling law?
S: It describes the length-tension relationship for muscle fibers.
T: Now can you define IS (which is also called cardiac contractility)?
S: The force with which the heart contracts.

T: Yes and IS is neurally determined. A change in IS will move shift the Frank-Starling curve along the axis.

Implementing a multistep dialogue like this is dificult in a system that generates and delivers sentences one at a time, when the plan must change if the student fails to answer a question.

9.2 Hints

From the begining of our work together, Michael and Rovick emphasized the importance of hints in tutoring. They told us that hints are an essential part of tutoring, and that they make frequent use of this strategy. They also suggested a rule of thumb for hinting. When the student gives a wrong answer, the system should hint. If the student still gets it wrong, the system should hint again. If the student gets it wrong the third time, the system should give the answer. So we tried to add hints to the generated dialogue, mostly beginning "Remember" or "Think about."

In the 1993 trials Michael and Rovick told us that the hints were terrible. We realized somewhere in this discussion that we had only recognized one type of hint – the reminder kind. We were missing half of the hints they were producing. At about this same time, Kurt Van Lehn came to visit and taught us a great deal about how to observe dialogues and ask questions. He arranged to interview the student while the session was in progress, something that we had never thought to try. This meant that the tutor had time to talk to an observer too. Gregory Hume, one of our Ph.D. students, seized the opportunity to observe Michael during a two-hour tutoring session. Michael described this student as a "live one," one who really responded to hints. The next student, he observed to Hume, was confused by hinting – and at this point, Michael stopped producing hints. We had not realized what a conscious process hinting is and we had not asked the right questions here. Hume chose hinting as a dissertation topic and started on a detailed study of hints and hinting strategies, which convinced us that hinting is a central issue in one-on-one tutoring (Hume et al., 1996). We found very little literature on this subject, perhaps because Grice (1968) disapproves of hinting.

Hume identified two broad hint categories, while analyzing a series of nine two-hour tutoring sessions. Hints either directly convey information to the student (ci-hints) or point to information (pt-hints). These two hint categories may be further broken down as shown in Figure 4.

Yujian Zhou has now implemented most of Hume's results on hinting, using APE operators. These operators make use of the student answer category (e.g., near-miss), the tutoring goal, and the local student assessment to determine the choice of hinting strategy from Hume's analysis. Examples are shown in Figure 5.

9.3 Discourse schemas and their implementation

Yuemei Zhang (1991) argued the need for schemas as high-level discourse plans in CIRCSIM-Tutor. Some of the schemas that she proposed wound up as discourse strategies in Version

CI (Convey Information) Hint Categories:

Summary and Question: This kind of hint is a review of previous material in the tutoring session followed by an explicit question.

Summary and Implied Question: Same as the previous category except that there is an implicit question in the hint.

Explanation and Question: This kind of hint is an explanation of information in the knowledge base followed by an explicit question.

Explanation and Implied Question: Same as the previous category except that there is an implicit question in the hint.

Explanation/Summary and Question: This kind of hint contains a summary of previous dialogue along with an explanation of information in the knowledg base followed by an explicit question.

Explanation/Summary and Implied Question: Same as the previous except that there is an implicit question in the hint.

PT (Point-To-information) Hint Categories:

Question, Explicit or Implied: In this hinting style, the tutor asks a question in hopes that the student can discover the solution to the problem using known information.

Reply to Student Question with a Question: In this category, the tutor tries to get the student to answer his or her own question.

Partial Acknowledgment (positive or negative): The student has partially answered the question, but has either forgotten something or incorrectly stated part of the answer.

Summary of Implications of Incorrect Student Response: This kind of hint uses the student's responses to show that following this logic contradicts some previously established rule or fact.

Figure 4. A Taxonomy of Hints (Hume et al., 1993, p. 566)

Near-Miss Example 1:
T: What is the determinant of Central Venous Pressure?
S: central blood volume
T: And what parameter in the prediction table determines Central
Blood Volume?
S: cardiac output
T: Right, CO determines CBV, therefore determines CVP.

Near-Miss Example 2:
T: What is the determinant of Central Venous Pressure?
S: cbv
T: And what parameter in the prediction table determines Central
Blood Volume?
S: is
T: No. Lets go back to the determinant of CVP.
Central Venous Pressure is determined by Cardiac Output.

Near-Miss Example 3:
T: What is the determinant of Central Venous Pressure?
S: blood volume
T: Blood Volume determines CVP. However, no change in blood
volume occurs in this situation.
S: co
T: Correct, Central Venous Pressure is determined by Cardiac Output.

Partially Correct Answer Case Example 1:
T: What are the determinants of Stroke Volume?
S: is
T: Right, Inotropic State is one determinant of Stroke Volume.
The other determinant represents preload.
What is the other determinant of Stroke Volume?
S: cvp
T: Right, Central Venous Pressure is one determinant of Stroke
Volume.
Stroke Volume is determined by Central Venous Pressure and
Inotropic state.

Incorrect Answer Case Example 1:
T: If afterload, i.e. MAP, decreased, which is what happened in
this case, what should SV be?
What is the correct value of Stroke Volume?
S: decrease
T: No, the value of Stroke Volume is not decreased.
Remember, if afterload decreased, the heart can pump blood out more
easily.
What is the correct value of Stroke Volume?
S: increased
T: Correct, the value of Stroke Volume is increased.

Mixed Answer Case Example 1:
T: What is the determinant of Central Venous Pressure?
S: blood volume blood pressure cardiac output
T: Right, CO is the one I am looking for. Blood Volume certainly
determines CVP. However, no change in blood volume occurs in this
situation. So, CO determines CVP.

Figure 5. Examples of Hints Given by Version 2 on 11/16/98.

2. But it was Reva Freedman (1996) who developed our current repertory of schemas as well as our current methodology for implementing them.

Freedman examined the discourse plans in our tutoring session transcripts and pointed out the wide variety of schemas used by Michael and Rovick. She also demonstrated the importance of the student answer in the choice of the expert response. She took scenarios written by Khuwaja for primary variable tutoring and developed families of schemas for them. She worked out many more scenarios herself to cover all our tutoring situations. See Figure 6 for an example. She then developed schemas for these scenarios and came up with a way to represent these schemas as plans.

The next step was to generalize these plans as planning operators. Freedman went on to build APE, the Atlas Planning Environment (2000a,b), while working with Kurt VanLehn on the CIRCLE project. She then expressed these planning operators as APE operators, which can now be executed by APE, as well.

9.4 Correction/acknowledgments

When Dr. Susan Chipman first used our system she commented on the fact that Version 2 was delivering acknowledgments much too often. Every time the student produced and answer the system responded with "Correct" or "Wrong." Human tutors do not do this.

Study of the transcripts showed that Michael and Rovick often combine negative acknowledgments and hints (Spitkowsky & Evens, 1993; Evens et al., 1993). To discover how these processes interact, we began by identifying the negative acknowledgments in the nine keyboard sessions used in our initial research on hints (K30-K38). Each of these sessions is two hours in length. In these sessions there are 197 negative acknowledgments and 194 hints. There are 125 cases where hints and negative acknowledgments are combined. Thus, if we look only at negative acknowledgments, out of the total of 197, 125 (63%) were combined with hints. If we look at hints, out of the total of 194, 125 (64%) are combined with negative acknowledgments. Hinting in a tutoring session can occur after a negative acknowledgment or in response to obvious student confusion or an explicit student initiative. Therefore, many hints were not associated with a negative acknowledgment. Equally, negative acknowledgments do not always lead into hints. This is because the tutor can give a negative response and follow it up with an explanation or just a simple statement of fact.

To discover how to avoid too many explicit acknowledgments, Stefan Brandle investigated our transcripts using Clark's theory of joint actions, and devised a number of rules for dropping acknowledgments. Underlying these rules are a couple of principles that we failed to grasp until Brandle's analysis. When a tutoring goal is satisfied, the tutor goes on to the next topic. Therefore, when the tutor changes the topic, the student can infer that the last answer was correct, but when the tutor continues on with the same topic, the student can infer that there is a problem. These are very general guidelines and a number of rules are needed to generate acknowledgments properly. For example, when the student has been doing badly or shows other evidence of confusion, the tutor will provide explicit positive acknowledgments.

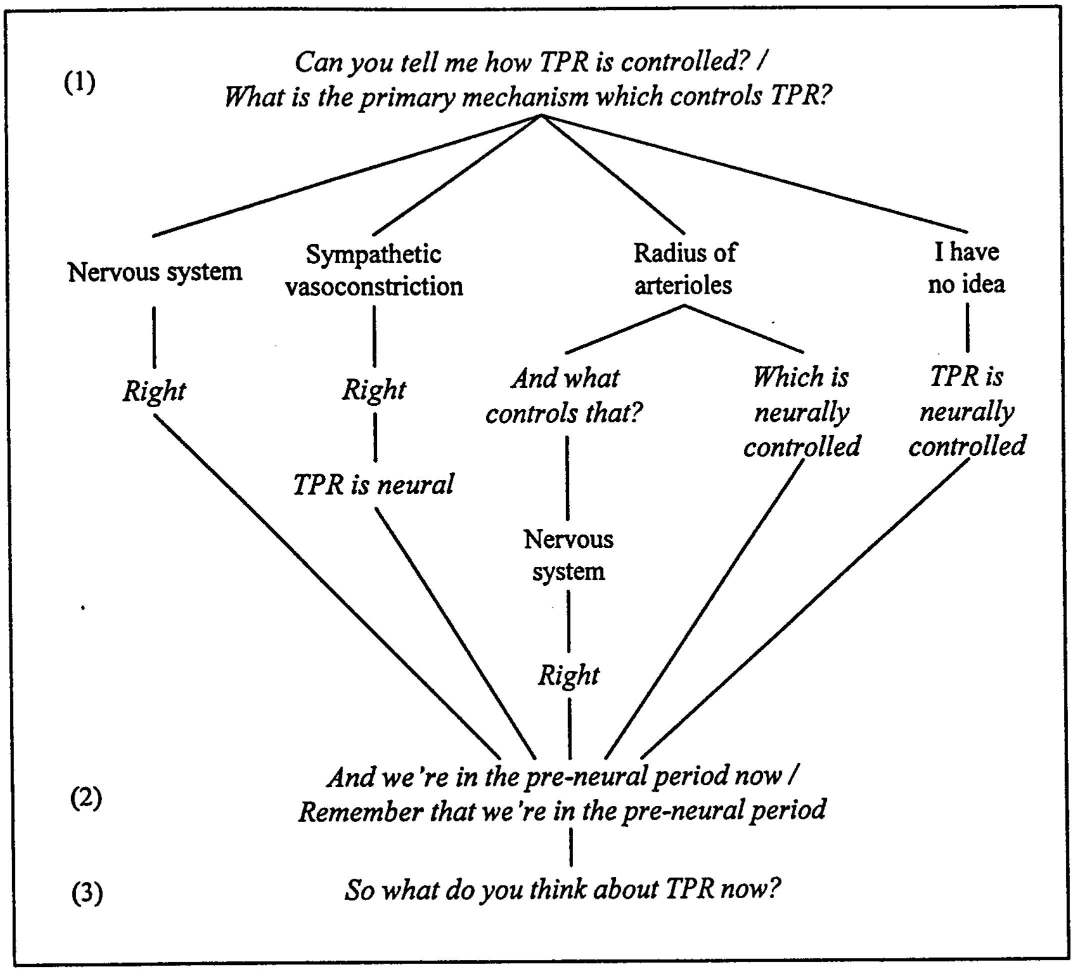

Figure 6. Schemas Developed by Reva Freedman (1996)

While our tutors use a large variety of negative acknowledgment strategies, they clearly use more explicit negative acknowledgments than the tutors studied by Fox (1993a,b). Where could we look for an explanation of these differences? There is certainly a difference in the social situations underlying these studies. In the case of the Fox study the tutors are graduate students hired to help undergraduates through a physics course. Our tutors are professors who are tutoring students taking a course from them that covers this same material. The tutors are also the employers in our situation. The educational situations are also very different. The students in our study are older than those Fox observed; they are learning material that is essential to their performance as professionals. Our tutors are also more experienced tutors, and we conjecture that experienced tutors are more likely to give explicit negative acknowledgments.

9.5 User-driven lexical choice

Yuemei Zhang (1991), who wrote our first generation program, remarked that verbs always occurred in antonym pairs in the trasncripts, so "go up" is paired with "go down," "increase" with "decrease," and "rise" with "fall." When Kumar Ramachandran set out to implement lexical choice in Version 2, he realized that the tutor's choice was based on the student's choice. If the student used acceptable language, the tutor would continue with the student's choice. He named this practice "user-driven lexical choice."

Ramachandran also argued that it was important for the machine tutor to make the student familiar with different terms for the same parameter. So he caused the system to cycle between the terms "Inotropic State," "IS," "Cardiac Contractility," and "CC." It was some unfortunate repercussions of this last decision that led us to the discovery of the need for turn planning. When the system used "Cardiac Contractility" later in a turn in which it had first used "CC," the student decided that the system was trying to hint, because people usually give the full name of a term first and then abbreviate it. Freedman (1996) looked at this example and some other bad turns and pointed out that lexical choice needs to be carried out in the context of a turn; sentence level planning is not adequate here.

We are now concerned particularly with the choice of discourse markers like "so" and "then," which can help us communicate the tutor's intent more clearly, and the choice of pronouns and other anaphora. Kim et al. (2000) combined corpus-based machine learning with traditional linguistic analyses to create rules for discourse marker selection.

We are using GenKit (Nyberg and Tomita, 1988) as an engine for surface realization in Version 3. We have found it to be both fast and flexible. Kim (2000) has written a first grammar for Version 3, but more work will be needed to expand it. The implementation of Zhou's (2000) rules for generating hints in Version 3 requires changes to the both the Student Modeler and the Turn Planner. The Turn Planner must pick up information about the previous question and the student answer from the Discourse History and then use the input from the Student Model to decide how to formulate the hint.

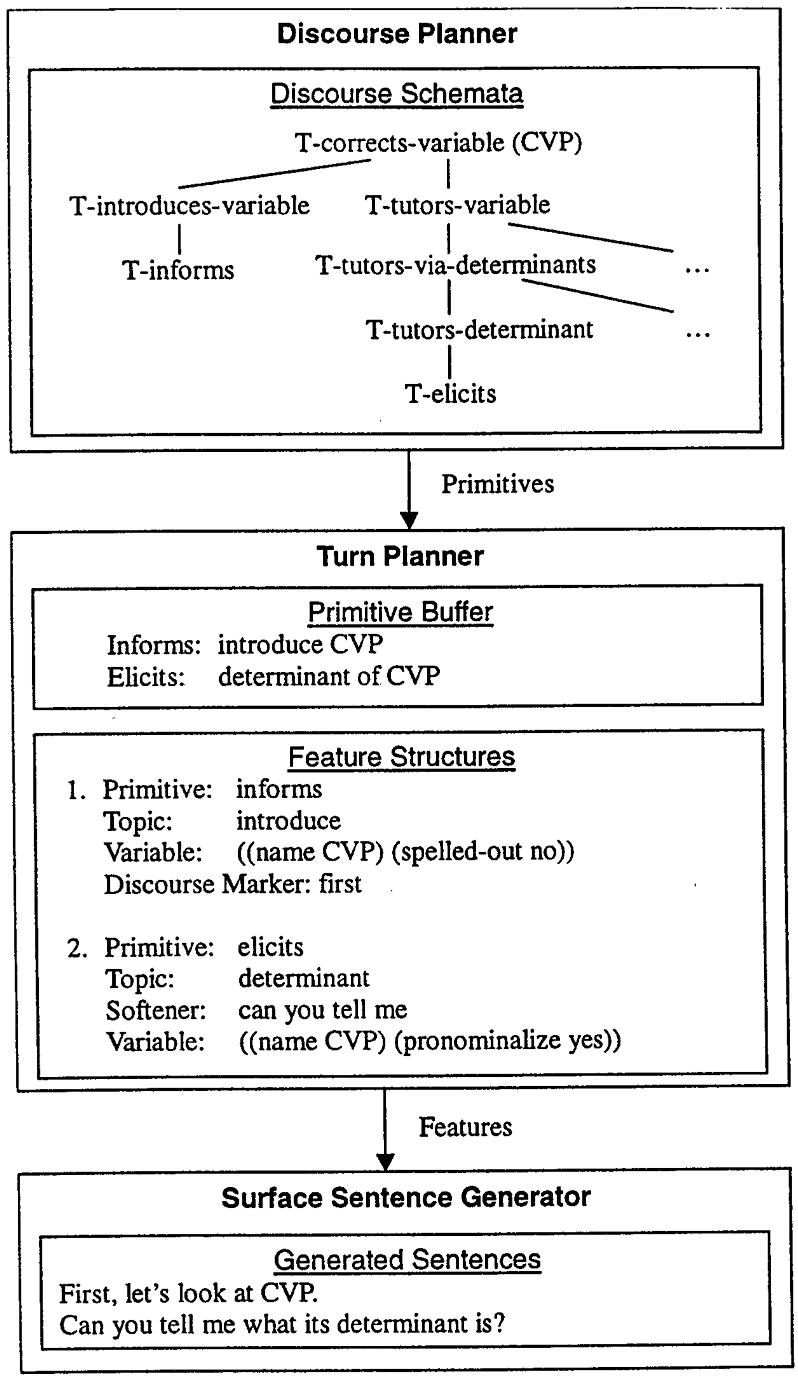

Figure 7. Levels of Dialogue Planning from (Yang, 2000b, p. 64)

– a realization of the dreams that began when ONR funded Carbonell (1970) and Collins thirty years ago in 1969.

Three important factors in our success, we are convinced, are the continued close collaboration between expert tutors and the implementers, the determination to model the system on human tutoring, and the opportunity for repeated trials with actual students at every stage of development.

What have we learned? Most studies of one-on-one tutoring have shown it to be remarkably effective for unmotivated, low-skilled teenagers. We have shown it to be just as effective for highly motivated, highly intelligent adult learners.

Graesser describes real tutors as using few, if any, of the sophisticated strategies described in the literature. As far as we can see, his tutors are all novice tutors. Our experts hint, show contradictions, ask diagnostic questions, structure the dialogue so that the students provide the answers whenever possible. We hope that our contrastive studies of novice vs. expert tutors may lead to new and more effective training for human tutors as well as better Intelligent Tutoring Systems.

Nothing can replace the insight gained by reading transcripts and talking to expert tutors, but the addition of discourse markup and machine learning has given us a powerful new way to confirm results, develop more effective rules, and give a scientific basis to this insight. The work of Hume and Freedman and Junghee Kim have been fundamental to our research.

Hume's studies of hints in tutoring have brought this important tutoring strategy to the notice of the world of ITS. Zhou has now implemented these discoveries in a principled and effective manner.

Hints are just one aspect of the multiturn discourse planning problems. Planning interactive explanations and summaries has been a major effort as well – at the discourse planning level, at the sentence planning level, at the lexical level. Our demonstration of the need for turn planning grew out of our efforts to do better lexical choice. We are continuing to work on discourse markers and anaphora. As our system grew we developed a need for curriculum planning, which Cho has satisfied.

Although the emphasis has been on language generation, interactive dialogue systems cannot function without the ability to process user input. Here our work on spelling correction and Glass' Information Extraction parser are our most important contributions.

All of this work required planning engines. The work of Woo on dynamic planning aroused a lot of interest when it was first published. The work of Freedman on dynamic, reactive planning is being widely used. We are continuing to write new plans today. Of course, we must share the honors here with the CIRCLE project. It makes us very happy to see some of these ideas continuing in the MURI research.

Any of the software described here is, of course, available to all who can use it. Please email us at evens@iit.edu or call 312-567-513.

11 Bibliography

The bibliography is divided into three parts: a list of references from other projects, a list of papers produced by the Circsim-Tutor project, and a list of theses.

11.1 List of References from outside the Circsim-Tutor Project

Ahlswede, T.E. (1985). A tool kit for lexicon building. *Proc. Association for Computational Linguistics*, Chicago, IL, 268-276.

Carberry, S. (1991). *Plan recognition in natural language dialogue.* MIT Press, Cambrige, MA.

Carbonell, J. (1970). AI in CAI: An artificial intelligence approach to computer-aided instruction. *IEEE Transactions on Man-Machine Systems,* 11(4): 190-202.

Clark, Herbert H. (1996). *Using language.* Cambridge UK: Cambridge University Press.

Conati, C., Gertner, A., VanLehn, K., and Druzdzel, M. (1997). On-line student modeling for coached problem solving using bayesian networks. *Proceedings of UM-97, Sixth International Conference of User Modeling,* 231-242.

Conlon, S.P., Dardaine, J., D'Souza, A., Evens, M., Haynes, S. Kim, J.S. and Strutz, R. (1994). The IIT lexical database: Dream and reality. In *Current Issues in Computational Linguistics: In Honour of Don Walker.* A. Zampolli, N. Calzolari, and M. Palmer, eds. Also *Linguistica Computazionale,* Vol. IX. Giardini Editori, Pisa, distributed in the US by Kluwer Academic Publishers. Norwell, MA. 201-225.

Conlon, S.P., Evens, M., Ahlswede, T., and Strutz, R. (1993). Developing a large lexical database for information retrieval, parsing, and text generation systems. *Journal of Information Processing and Management,* 29(4): 415-431.

Fox, B. (1993a). Correction in tutoring. *Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society,* Boulder, CO. 121-126.

Fox, B. (1993b). *The Human Tutoring Dialogue Project.* Erlbaum, Hillsdale, NJ.

Freedman, R. (2000a). Plan-based dialogue management in a physics tutor. *Proceedings of the Sixth Applied Natural Language Processing Conference.* Seattle, WA.

Freedman, R. (2000b). Using a reactive planner as the basis for a dialogue agent. *Proceedings of FLAIRS 2000,* Orlando, FL.

Freedman, R., Rosé, C.P., Ringenberg, M., & VanLehn, K. (2000). ITS Tools for Natural Language Dialogue. *ITS 2000,* Montreal, 433-442.

Gertner, A., Conati, C., and VanLehn, K. (1998). Procedural help in Andes: Generating hints using a Bayesian network student model. *Proc. AAAI.* 106-111.

Graesser, A.C. (1993). Dialogue patterns and feedback mechanisms during naturalistic tutoring. *Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society,* Boulder, CO. 127-130.

Graesser, A.C., Person, N.K., & Huber, J. (1993). Question asking during tutoring and in the design of educational software. In *Cognitive Science Foundations of Instruction,* Rabinowitz, M. ed., Erlbaum, Hillsdale, NJ. 149-172.

Graesser, A.C., Franklin, S., Wiemer-Hastings, P. (1998). Simulating smooth tutorial dialogue with pedagogical value. *Proc. FLAIRS 98.* Sanibel, Island, FL. 163-167.

Graesser, A.C. (1993a). *Questioning mechanisms during tutoring conversation and human-computer interaction.* Technical Report R&T 4422576 of the Cognitive Science Program, Office of Naval Research.

Graesser, A.C. (1993b). Dialogue patterns and feedback mechanisms during naturalistic tutoring. *Proc. COGSCI '93,* Boulder, CO. 126-130.

Graesser, A.C., Person, N.K., & Magliano, J.P. (1995). Collaborative dialogue patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology,* 9, 495-522.

Grice, H. Paul. (1968). Logic and conversation. In Peter Cole & Jerry Morgan, eds. (1975). *Syntax and Semantics,* reprinted from *Studies in the Way of Words,* Harvard University Press, (1968) 41-58.

Hovy, Eduard H. (1988). Planning coherent multisentential text. *Proceedings of the 26th Annual Meeting of the ACL.* 163-169.

Kaplan, R. M. and Bresnan, J. (1982). Lexical Functional Grammar: A formal system for grammatical representation. In J. Bresnan, (Ed.). *The mental representation of grammatical relations.* Cambridge, MA: MIT Press.

Lambert, L., & Carberry, S. (1992). Modeling negotiation subdialogues. *30th Annual Meeting of the ACL.* 193-200.

Lesgold, A. (1992). Going from intelligent tutoring systems to tools for learning. In C. Frasson, G. Gauthier, & G. I. McCalla (Eds.), *Intelligent Tutoring Systems (Proceedings of the Second International Conference, ITS '92, Montreal, CANADA).* Springer-Verlag, Berlin.

Mann, W., & Thompson, S.A. (1986). Relational propositions in discourse. *Discourse Processes,* 9, 57-90.

Mann, W., & Thompson, S.A. (1987). Rhetorical structure theory: a theory of text organization. Technical Report ISI/RS-87-190. Marina del Rey: University of Southern California/Information Sciences Institute. Reprinted in Polanyi, Livia, ed.(1987). *The Structure of Discourse.* Norwood, NJ: Ablex.

Mann, W., & Thompson, S.A., eds. (1992). *Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text.* John Benjamins, Philadelphia, PA.

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: a comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences,* 2. 277-305.

Moore, J.D. (1993). What makes human explanations effective? *Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society,* Boulder, CO. 131-136.

Moore, J.D., & Paris, C.L. (1989). Planning text for advisory dialogues. *Proceedings of the 27th Annual Meeting of the ACL.* 203-211.

Moore, J.D., & Paris, C.L. (1993). Planning text for advisory dialogues: capturing intentional and rhetorical structure. *Computational Linguistics,* 19(4), 651-695.

Moore, J.D., & Pollack, M. (1992). A problem for RST: the need for multi-level discourse analysis. *Computational Linguistics,* 18(4), December, 1992, 537-544.

Nyberg, Eric H. and Masaru Tomita. 1988. *Generation Kit and Transformation Kit Version 3.2 User's Manual,* Report CMU-CMT-88-Memo from the Center for Machine Translation at Carnegie-Mellon University.

Roche, E., & Schabes, Y. (1997). *Finite-State Language Processing.* MIT Press, Cambridge, MA.

Rovick, A.A., & Brenner, L. (1983). HEARTSIM: A cardiovascular simulation with didactic feedback. *Physiologist,* 26(4), 236-239.

Smith, R.W., Hipp, D.R., & Biermann, A. (1992). A dialog control algorithm and its performance. In *Third Conference on Applied Natural Language Processing,* ACL. 9-16.

Thompson, B.H. (1980). Linguistic analysis of natural language communication with computers. *COLING 80,* Tokyo, 190-201.

Woolf, B. (1984). *Context-dependent planning in a machine tutor.* Ph.D. diss., Dept. of Computer and Information Science, University of Massachusetts at Amherst. COINS Technical Report 84-21.

Woolf, B. and Murray, T. (1994). Using machine learning to advise a student model. Greer, J.E. and McCalla, G.I., Eds., *Student modelling: The key to individualized knowledge-based instruction,* Berlin: Springer-Verlag, 127-146.

Young, R.M. (1994). *A Developer's Guide to the Longbow Discours Planning System.* University of Pittsburgh Intelligent Systems Program Technical Report 94-4.

Zhang, Y., Evens, M., Michael, J., & Rovick, A. (1987). Knowledge compiler for an expert physiology tutor. *Proc. ESD/SMI Conference on Expert Systems,* Dearborn, MI, June, 1987, 153-169.

List of Papers Produced by the Circsim-Tutor Project.

Abbas, H. & and Evens, M. (2000). Domain knowledge base for an intelligent tutoring system: CIRCSIM-Tutor. *CATA,* New Orleans, March 31, 2000. 338-343.

Brandle, S., & Evens, M. (1997a). Acknowledgments in tutorial dialogue. *Proceedings of MAICS '97.* Dayton, OH, June. 13-18.

Brandle, S., & Evens, M. (1997b). Organizing acknowledgments in tutorial dialogue. *Proceedings of the Cognitive Science Conference,* August, Stanford, CA. 872.

Brandle, S. & Evens, M. (1988). Categorizing acknowledgements in tutorial dialogue. *Proceedings of CogSci '98,* Madison, Wisconsin.

Chang, R.C., & Evens, M. (1991). Developing a sublanguage grammar and lexicon using lexical-functional grammar. *Proceedings of the Third Midwest Artificial Intelligence and Cognitive Science Society Conference,* Carbondale, IL. 46-51.

Chang, R.C., Evens, M., Rovick, A.A., & Michael, J.A. (1992). Surface generation in a tutorial dialogue based on analysis of human tutoring sessions. *Fifth IEEE Symposium on Computer-Based Medical Systems,* Durham, NC, June 14-17. 554-561.

Chang, R.C., Evens, M., Michael, J.A., & Rovick, A.A. (1994). Surface generation in tutorial dialogues based in a sublanguage study. *Proc. ICAST'94,* Naperville, IL. March, 1994. 113-119.

Cho, B.I., Michael, J. A., Rovick, A. A., & Evens, M. W. (1999). A curriculum planning model for an intelligent tutoring system. *Proceedings of the 12th Florida Artificial Intelligence Symposium (FLAIRS-99),* Orlando, FL. 197-201.

Cho, B.I., Michael, J., Rovick, A., & Evens, M.W. (2000). An Analysis of multiple tutoring protocols. *Proc. Intelligent Tutoring Systems, ITS 2000.* Montreal, PQ. 212-221.

Dardaine, J. (1992). Case frames for a lexical database. *Proceedings of the Third Midwest Artificial Intelligence and Cognitive Science Society Conference,* Carbondale, IL. 102-106.

Elmi, M., & Evens, M. (1993). An efficient natural language parsing method. *Proceedings of Midwest Artificial Intelligence and Cognitive Science Conference,* Chesterton, IN. 6-10.

Elmi, M., & Evens, M. (1998). Spelling correction using context. *Proceedings of COLING 98,* Montreal, Canada. 360-364.

Evens, M., Spitkovsky, J., Boyle, P., Michael, J.A. & Rovick, A.A. (1993). Synthesizing tutorial dialogues. *Proceedings of CogSci '93.* Boulder, CO, June. 137-142.

Freedman, R. (1995). Using pedagogical knowledge to structure text generation in an intelligent tutoring system. *Proceedings of the Midwest Artificial Intelligence and Cognitive Science Conference, MAICS '95,* Carbondale, IL, April. 48-52.

Freedman, R. (1996a). Using a text planner to model the behavior of human tutors in an ITS, *Proceedings of the 1996 Midwest Artificial Intelligence and Cognitive Science Society Conference,* Bloomington, IN. http://www.cs.indiana.edu/event /maics96/Proceedings/Freedman /freedman.html

Freedman, R. (1996b). Using tutoring patterns to generate more cohesive text in an intelligent tutoring system. *Proceedings of International Conference on Learning Systems (ICLS-96),* Evanston, IL. 75-82.

Freedman, R. (1997a). Degrees of mixed-initiative interaction in an intelligent tutoring system. *Computational Models for Mixed Initiative Interaction, AAAI Spring Symposium,* Stanford University, March 24-26, 44-49.

Freedman, R. (1997b). Representing communicative action in a dialogue-based intelligent tutoring system. *AAAI Fall Symposium on Communicative Action in Humans and Machines.*

Freedman, R. & Evens, M. (1996a). Generating and revising hierarchical multi-turn text plans in an ITS. *Proceedings of the Third International Conference on Intelligent Tutoring Systems (ITS '96),* Montreal, Canada. 632-640.

Freedman, R. & Evens, M. (1996b). Realistic limitations in natural language processing for an intelligent tutoring system. *Proceedings of 20th Annual Cognitive Science Conference*, La Jolla, CA.

Freedman, R. & Evens, M. (1997). The use of multiple knowledge types in an intelligent tutoring system. *Proceedings of the Cognitive Science Conference,* Stanford, CA. 920.

Freedman, R., Zhou, Y., Kim, J.H., Glass, M., & Evens. M. (1998a). SGML-based markup as a step toward improving knowledge acquisition for text generation, *AAAI Spring Symposium on Applying Machine Learning to Discourse Processing.*

Freedman, R., Zhou, Y., Glass, M., Kim, J.H., & Evens. M. (1998b). Using rule induction to assist in rule construction for a natural-language based intelligent tutoring system. *Proceedings of 20th Annual Cognitive Science Conference,* Madison, WI, August. 362-367.

Freedman, R., Brandle, S., Glass, M., Kim, J.H., Zhou, Y., & Evens, M. (1998c). System demonstration: Content planning as the basis for an intelligent tutoring system. *International Conference on Natural Language Generation.* Niagara-on-the-Lake, Ontario, CA, August, 1998. 280-283.

Glass, M. (1997). Some phenomena handled by the Circsim-Tutor Version 3 input understander. *Proceedings of the Tenth International Florida Artificial Intelligence Research Symposium,* Daytona Beach, FL. 21-25.

Glass, M. (2000). Processing language input in the CIRCSIM-Tutor intelligent tutoring system. *Proceedings of the AAAI Fall Symposium on Dialogue Systems.*

Glass, M., & Evens, M. (1996). Goals for the CIRCSIM-Tutor input understander. *Proceedings of MAICS96,* Bloomington, IN. http://www.cs.indiana.edu/event/maics96/Proceedings/glass.html

Glass, M., Kim, J.H., Evens, M., Michael, J.A., & Rovick, A.A. (1999). Novice vs. expert tutors: A comparison of style. *Proceedings of MAICS 99,* Bloomington, IN, April 24. 43-49.

Hume, G. (1992). A dynamic student model in a cardiovascular intelligent tutoring system, *Proceedings of the Fifth CBMS,* Durham, NC, June. 370-377.

Hume, G., & Evens, M. (1992). Student modeling and the classification of errors in an intelligent cardiovascular tutoring system, *Proceedings of the Fourth Midwest Artificial Intelligence and Cognitive Science Conference*, Starved Rock, IL, May. 52-56.

Hume, G., Evens, M., Rovick, A.A., & Michael, J. (1993). The use of hints as a tutoring tactic. *Proceedings of CogSci '93.* Boulder, CO, June. 563-568.

Hume, G., Michael, J.A., Rovick, A.A., & Evens, M. (1995). Controlling active learning:

How tutors decide when to generate hints. *Proceedings of FLAIRS '95.* Melbourne Beach, FL, April. 157-161.

Hume, G., Michael, J.A., & Rovick, A.A. (1996) The student model: from text-based to multimedia tutoring systems. Online proceedings of the 8th Midwest Artificial Intelligence and Cognitive Society Conference. URL: http://www.cs.indiana.edu/event/maics96/Proceedings/hume.html

Hume, G., Michael, J.A., Rovick, A.A., & Evens, M. (1996a). Hinting as a tactic in one-on-one tutoring. *Journal of the Learning Sciences,* Vol. 5, No. 1, 23-47.

Hume, G., Michael, J.A., Rovick, A.A., & Evens, M. (1996b). Student responses and follow up tutorial tactics in an ITS. *Proceedings of the 1996 Florida Artificial Intelligence Research Symposium.* May 20-22, Key West, FL, 168-172.

Hume, G., Michael, J.A., Rovick, A.A., & Evens, M. (1996c) The use of hints by human and computer tutors: the consequences of the tutoring protocol. Proceedings of the 2nd International Conference on the Learning Sciences. Evanston, IL, 135-142.

Jeong, I., Evens, M., & Kim, Y.K. (1998a). Tool for knowledge acquisition and knowledge visualization, *Proceedings of FLAIRS-98.* 173-177.

Jeong, I., Evens, M. & Kim, Y.K. (1998b). Tools for building concept maps. *Korea Telecom Journal,* Vol. 3, Number 1, (December, 1998), 11-21.

Khuwaja, R.A., & Patel, V. (1996). A model of tutoring based on the behavior of effective human tutors. *Proceedings of the Third International Conference on Intelligent Tutoring Systems (ITS '96)*, Montreal, Canada, 130-138.

Khuwaja, R., Evens, M., Rovick, A.A. & Michael, J. (1992). Knowledge representation for an intelligent tutoring system based on a multilevel causal model. *Proceedings of ITS '92*, Montreal, June. 217-224.

Khuwaja, R.A., Evens, M., Rovick, A.A., Michael, J. (1993). Building the domain expert for a cardiovascular physiology tutor. *Proceedings of the Sixth Annual IEEE Symposium on CBMS,* Ann Arbor, MI, June 13-16. 106-11.

Khuwaja, R.A., Evens, M., Rovick, A.A., Michael, J.A. (1994a). Architecture of CIRCSIM-TUTOR (v.3): A smart cardiovascular physiology tutor. *Proc. CBMS94,* Winston-Salem, NC, June 10-11. 158-163.

Khuwaja, R.A., Rovick, A.A., Michael, J.A. & Evens, M. (1994b). A tale of three tutoring protocols: The implications for intelligent tutoring systems. *Intelligent Systems: Proceedings of Golden West,* Las Vegas, NV, June 9-12. 109-118.

Kim, J.H. (1999). *The SGML Markup Manual for Circsim-Tutor.* Technical Report, Computer Science Department, Illinois Institute of Technology, Chicago, IL 60616.

Kim, J.H., Freedman, R., & Evens, M. (1998). Relationship between tutorial goals and sentence structure in a corpus of tutoring transcripts, *Ninth Midwest Artificial Intelligence and Cognitive Science Conference,* Dayton, OH, AAAI Press. 124-131.

Kim, J.H., Freedman, R., & Evens, M. (1998). Responding to unexpected student utterances in Circsim-Tutor v.3: Analysis of transcripts. *FLAIRS-98. Florida Artificial Intelligence Research Symposium,* Sanibel Island, FL. 153-157.

Kim, J.H., Glass, M., Freedman, R., & Evens, M. (2000). Learning the use of discourse markers in tutorial dialogue for an intelligent tutoring dystem. *Proc. Cognitive Science 2000.* Philadelphia, PA. 262-267.

Kim, N., Evens, M., Michael, J.A., & Rovick, A.A. (1989). CIRCSIM-Tutor: An intelligent tutoring system for circulatory physiology. In H. Maurer, ed. *Computer assisted learning.* Springer-Verlag, Berlin. 254-266.

Lee, Y.H., Evens, M., Michael, J.A., & Rovick, A.A. (1990). IFIHS: Ill-formed input handling system. *Proceedings of the Second Midwest Artificial Intelligence and Cognitive Science Conference.* Carbondale, IL, March 30-April 1. 93-97.

Lee, Y.H., Evens, M., Michael, J.A., & Rovick, A.A. (1989/1991). Spelling correction for an intelligent tutoring system. *Proceedings of the First Great Lakes Computer Science Conference*, Kalamazoo, October 18-20, 1989. Lecture Notes in Computer Science, 507, Springer, New York, 1991, 77-83.

Lee, Y.H., & Evens, M.W. (1992). Ill-formed input handling system for an intelligent tutoring system. *The Second Pacific Rim International Conference on Artificial Intelligence.* Seoul, Korea, September 15-18. 354-360.

Lee, Y.H., & Evens, M.W. (1998). Natural language interface for an expert system. *Expert Systems (International Journal of Knowledge Engineering).* November, Vol. 15, No. 4, 233-239.

Li, J., Rovick, A., & Michael, J. A. (1992a). ABASE - a computer program that teaches physiological acid base regulation. In I. Tomek (Ed.)., *Computer assisted learning (Proceedings of the 4th International Conference, ICCAL '92, Wolfville, NS, CANADA).* Berlin: Springer-Verlag. 380-390.

Li, J., Seu, J., Evens, M., Michael, J.A., & Rovick, A.A. (1992b). Computer Dialogue System (CDS): A system for capturing computer-mediated dialogues. *Behavior Research Methods, Instruments, and Computers (Journal of the Psychonomic Society).* Vol. 24, No. 4, 535-540.

Mayer, G., Yamamoto, C., Evens, M., & Michael, J. (1989). Constructing a knowledge base from a natural language text. *Proceedings of the 2nd Annual IEEE Symposium on Computer Based Medical Systems*, Minneapolis, June 25-27. 98-107.

Michael, J. A., & Rovick, A. A. (1991a). Evaluating the effectiveness of teaching software. *American Association for the Advancement of Science Annual Meeting.* Washington, DC.

Michael, J. A., & Rovick, A. A. (1991b). Does use of a CBE program assist students to learn? *Federation of American Societies for Experimental Biology Annual Meeting.* Atlanta, GA.

Michael, J. A., & Rovick, A. A. (1993). The effectiveness of human tutoring sessions. Unpublished paper, Department of Physiology, Rush Medical College, Chicago, IL.

Michael, J. A., & Rovick, A. A. (1996). Results of the pretests and post-tests from the novice tutoring sessions. Unpublished paper, Department of Physiology, Rush Medical College, Chicago, IL.

Michael, J. A., Rovick, A. A., Evens, M., & Kim, N. (1990.) A smart tutor based on a qualitative causal model. *Proceedings of the AAAI Spring Symposium on Knowledge-Based Environments for Learning and Teaching,* Stanford, CA, March 27-29, 112-117.

Michael, J.A., Rovick, A.A., Evens, M.W., Shim, L., Woo, C., & Kim, N. (1992). The uses of multiple student inputs in modeling and lesson planning in CAI and ICAI programs. *Computer Assisted Learning, Proc. ICCAL Conference*, I. Tomek (ed.), Wolfville, Nova Scotia, June, 1992, 441-452.

Michael, J. , Rovick, A., & Evens, M. (1994). Circsim-tutor: a smart tutor/learning environment based on a qualitative, casual model. *Proceedings of the International Conference on Computer- Assisted Education and Training in Developing Countries,* Midrand, Johannesburg, South Africa. 173-178.

Ramachandran, K. & Evens, M. 1995. Lexical choice for an intelligent tutoring system. *Proceedings of MAICSS '95,* Carbondale, IL, April, 53-57.

Rovick, A. A., & Michael, J. A. (1992). The prediction table: A tool for assessing students' knowledge. *American Journal of Physiology, 263 (Advances in Physiology Education, 8),* S33-S36.

Sanders, G., Evens, M., Hume, G., Rovick, A.A., & Michael, J.A. (1992). An analysis of how students take the initiative in keyboard-to-keyboard dialogues in a fixed domain. *Proceedings of the Cognitive Science Conference*, Bloomington, August. 1086-1091.

Seu, J., Evens, M., Michael, J.A., & Rovick, A.A. (1991a). Understanding ill-formed input to an intelligent tutoring system in an LFG framework. *Proceedings of the Third Midwest Artificial Intelligence and Cognitive Science Conference.* Carbondale, April. 36-40.

Seu, J., Chang, R-C., Li, J., Evens, M., Michael, J.A. & Rovick, A.A. (1991b). Language differences in face-to-face and keyboard- to-keyboard sessions. *Proceedings of the Cognitive Science Conference*, Chicago, IL, August. 576-580.

Shah, F. & Evens, M. (1997). Student initiatives and tutor responses in a medical tutoring system. In *Computational Models for Mixed Initiative Interaction.* AAAI Spring Symposium, Stanford, CA, March 24-26, 138-144.

Shim, L., Evens, M., Rovick, A.A., & Michael, J.A. (1990). Student modelling issues in intelligent tutoring systems. *Proceedings of the Third University of New Brunswick Artificial Intelligence Workshop*, Fredericton, NB, October. 127-136.

Shim, L., Evens, M., Michael, J.A., & Rovick, A.A. (1991a). Student modeling for tutoring causal relationships. *Proceedings of the Third Midwest Artificial Intelligence and Cognitive Science Conference.* Carbondale, IL, April, 1991, 26-30.

Shim, L., Evens, M., Michael, J.A., & Rovick, A.A. (1991b). Effective cognitive modeliing in an intelligent tutoring system for cardiovascular physiology. *Proceedings of the Fourth Annual IEEE Symposium on Computer Based Medical Systems*, Baltimore, MD, May, 1991, 338-345.

Spitkovsky, J. & Evens, M. (1993). Negative acknowledgements in natural language tutoring. *Proceedings of MAICSS '93,* Chesterton, Indiana, April 18-19. 41-45.

Sukthankar, S., Ramachandran, K., Evens, M., Rovick, A.A., & Michael, J. (1993). Graphical user interface with domain visualization for an intelligent medical tutoring system. *Proceedings of the Sixth Annual Symposium on CBMS,* Ann Arbor, June 13-16. 189-193.

Woo, Chong Woo. 1992. *A multi-level dynamic instuctional planner for an intelligent tutoring system.* ONR Technical Report.

Woo, C.W., Evens, M., Michael, J.A., & Rovick, A.A. (1991a). Instructional planning for an intelligent medical tutoring system. *Proceedings of the Third Midwest Artificial Intelligence and Cognitive Science Conference.* Carbondale, IL, April, 1991, 31-35.

Woo, C.W., Evens, M., Michael, J.A., & Rovick, A.A. (1991b). Dynamic planning in an intelligent cardiovascular tutoring system. *Proceedings of the Fourth Annual IEEE Symposium on Computer Based Medical Systems*, Baltimore, May, 1991, 226-233.

Woo, C., Evens, M., Michael, J.A., & Rovick, A.A. (1991c). Planning in an intelligent tutoring system. *Poster Session of the International Conference on the Learning Sciences,* Evanston, IL.

Yang, F.J., Kim, J.H., Glass, M., & Evens, M. (2000a). Lexical issues in the tutoring schemata of CIRCSIM-Tutor: Analysis of variable references and discourse markers. *Proc. Human Interfaces to Complex Systems.* Beckman Institute, Urbana, April. 26-31.

Yang, F.J., Kim, J.H., Glass, M. & Evens, M. (2000b). Turn Planning in CIRCSIM-Tutor. *Proc. FLAIRS,* May. 60-64.

Zhang, Y., Evens, M., Michael, J.A. & Rovick, A.A. (1990). Extending a knowledge base to support explanations. *Proceedings of the Third IEEE Conference on Computer-Based Medical Systems*, Chapel Hill, NC, June 4-6. 259-266.

Zhou, Y., Freedman, R. K., Glass, M., Michael, J., A., Rovick, A. A., & Evens, M. W. (1999a). What Should the Tutor Do When the Student Cannot Answer a Question? Proceedings of the 12th Florida Artificial Intelligence Symposium (FLAIRS-99), Orlando, FL. 187-191.

Zhou, Y., Freedman, R., Glass, M., Michael, J.A., Rovick, A.A., & Evens, M. (1999b). Delivering hints in a dialogue-based ITS. *Proceedings of AAAI.* Orlando, FL. 128-134.

Zhou, Y., & Evens, M. (1999). A practical student model in an intelligent tutoring system. *Proc. 11th IEEE International Conference on Tools with Artificial Intelligence.* Chicago, November 9, 1999. 13-18.

Ph.D. Theses Related to the Circsim-Tutor Project

Nakhoon Kim, *An Intelligent Tutoring System for Physiology.* December, 1989.

Yoon Hee Lee, *Handling Ill-Formed Natural Language Input for an Intelligent Tutoring System.* August, 1990.

Leemseop Shim, *Student Modeling for an Intelligent Tutoring System Based on the Analysis of Human Tutoring Sessions.* August, 1991.

Chong Woo Woo, *Instructional Planning in an Intelligent Tutoring System: Combining Global Lesson Plans with Local Discourse Control.* December, 1991.

Yuemei Zhang, *Knowledge-Based Discourse Generation for an Intelligent Tutoring System.* December, 1991.

Jai Hyun Seu, *The Development of an Input Understander for an Intelligent Tutoring System Based on a Sublanguage Study.* May, 1992.

Ru-Charn Chang, *Surface Level Generation of Tutorial Dialogue Using a Specially Developed Lexical Functional Grammar and Lexicon.* August, 1992.

Glenn Mayer, *Creating a Structured Knowlege Base by Parsing Natural Language Text.* December, 1992.

M. Ali Elmi, *A Natural Language Parser with Interleaved Spelling Correction Supporting Lexical Functional Grammar and Ill-Formed Input.* December, 1994.

Ramzan Ali Khuwaja, *A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain.* December, 1994.

Gregory Hume, *Using Student Modeling to Determine How and When to Hint in an Intelligent Tutoring System.* May, 1995.

Gregory Sanders, *Generation of Explanations and Multi-Turn Discourse Structures in Tutorial Dialogue Based on Transcript Analysis.* July, 1995.

Joanne Dardaine, *Towards the Semiautomatic Generation of IITROLE: A Case Model Incorporating Syntactic, Semantic, and Pragmatic Information.* December, 1995.

Reva Freedman, *Interaction of Discourse Planning, Instructional Planning, and Dialogue Management in an Interactive Tutoring System,* Department of EECS, Northwestern University, Evanston, IL, December, 1996.

Farhana Shah, *Recognizing and Responding to Student Plans in an Intelligent Tutoring System: CIRCSIM-Tutor.* July, 1997.

Stefan Brandle, *Using Joint Actions to Explain Acknowledgments in Tutorial Discourse: Application to Intelligent Tutoring Systems.* May, 1998.

Hasan Abbas, *Designing a New Domain Knowledge Base for an Intelligent Tutoring System, Circsim-Tutor V.3.* December, 1998.

Michael Glass, *Broadening Input Understanding in a Language-Based Intelligent Tutoring System.* May, 1999.

Junghee Kim, *Natural Language Analysis and Generation for Tutorial Dialogue.* May, 2000.

Yujian Zhou, *Building a New Student Model to Support Adaptive Tutoring in a Natural Language Dialogue System.* May, 2000.

Byung-In Cho, *Dynamic Planning Models to Support Curriculum Planning and Multiple Tutoring Protocols in Intelligent Tutoring Systems,* July, 2000.

Feng-Jen Yang, *Turn Planning and Lexical Choice in a Natural Language Dialogue-Based Intelligent Tutoring System,* expected July, 2001.

D. Bruce Mills, *Creating Discourse Plans and the Supporting Software Architecture for Tutoring Using Freedman's ATLAS Planning Environment,* expected July, 2002.

Sun M. Li, *Portability Issues in Intelligent Tutoring Systems,* expected, July, 2002.

M.S. Thesis Related to the Circsim-Tutor Project

Kumar Ramachandran. *Lexical Choice in Natural Language Text Generation for an Intelligent Medical Tutoring System,* May, 1994.